

Assessment of a Novel Technique for Indexing Real World Temporal Cases

Mykola Galushka, David Patterson

Northern Ireland Knowledge Engineering Laboratory (NIKEL)
Faculty of Engineering
University of Ulster
Jordanstown
N. Ireland
mg.galushka, wd.patterson @ulster.ac.uk

Abstract

Temporal features are an important characteristic of real world data, yet often they are ignored within the context of case-base reasoning (CBR) systems. This is largely due to the difficulties of handling and maintaining temporal knowledge effectively. We present a novel, domain independent approach, called D-HS^T, designed for case retrieval in CBR, we assess its limitations and evaluate modifications designed to address these shortcomings. We show the approaches to be both efficient and competent when indexing real world case-bases, compared to the commonly used R-tree indexing technique.

Introduction

Case-based reasoning (CBR) has been successfully applied in a wide variety of domains. Most systems focus on time invariant attributes and either ignore temporal attributes or oversimplify them [1]. This is largely due to the difficulty of satisfactorily handling the CBR processes, such as, similarity determination, indexing, adaptation and knowledge maintenance, with time related data. However time is an important and pervasive concept in the real world [2] and therefore by default, important to many of the domains CBR can be applied to. Examples of these domains include, reasoning with real time data obtained from sensors (automatic control systems) or medical diagnostics (monitoring a patients vital signals). The growing importance of handling temporal data is clearly seen by observing the recent increase in the volume of research on temporal CBR (T-CBR) systems [3].

Most research into indexing temporal data has been carried out in the context of temporal data mining where the majority of techniques fall into a two-level framework, with level one involving feature extraction (FE) and level two indexing. It can be seen that the majority of FE techniques focus on efficient dimensionality reduction approaches

such as: Discrete Fourier Transformation (DFT) [4], Discrete Wavelet Transformation [5], different variations of Piecewise Approximation [6] and Languages for Shape Representation [7]. The number of actual indexing techniques employed by researchers is limited [8]. The most popular and widely used being the R-tree [9] indexing structure and its variations.

As most temporally orientated domains are likely to have a mixture of both temporal and non-temporal attributes it is important that all the CBR processes can handle both types interchangeably. It is easy to apply k-NN to retrieve similar cases based on both attribute types, however this approach is very inefficient if the case-base grows in size or there are more than a couple of temporal dimensions (after feature extraction it is not uncommon for each temporal attribute to be transformed into 10 frequency features). R-trees have been used to create indexes for T-CBR to overcome the efficiency limitation of k-NN, but it has a dimensionality limitation. According to [8] an R-trees efficiency deteriorates when dimensionality increases beyond 20, which is approximately enough to index only one or two temporal attributes (depending on the feature extraction method applied).

A novel, domain independent, indexing technique called D-HS^T (Discretized Highest Similarity^{Temporal}), which can effectively index both temporal and non temporal attributes within a case structure, is evaluated in this paper. We assess its limitations and as such propose 2 modifications of the algorithm designed to improve its competency. Previously D-HS has been shown to be effective at indexing cases with time invariant attributes [10]. In [11] an extension was proposed for handling cases with time-series (TS) attributes and a very limited initial analysis of the technique using synthetic cases provided. Here we present a more thorough empirical evaluation of D-HS^T with real world temporal cases, propose improvements to it and benchmark them all against R-trees. We show it to be comparable to R-trees in terms of accuracy but more efficient, depending on the variation used. Due to space limitations

we only focus here on indexing cases consisting wholly of temporal attributes and ignore hybrid cases consisting of both temporal and non-temporal attributes.

The next section outlines the methodology of the techniques, which is followed by the experimental set up and results of the comparison between D-HS^T and R-tree. Finally a conclusion and future work is proposed.

Methodology

Firstly, due to space restrictions, here we only provide a high level methodology overview of both the D-HS and D-HS^T approaches. For more information interested readers are encouraged to read [10] and [11] respectively. We then describe a number of improvements to D-HS^T developed as a result of recent work. D-HS is based on a matrix-like indexing structure where columns represent indexing spaces for attributes, and rows represent their segmentations. Each attribute is represented in the matrix as a single column. The number of rows (segments) for each attribute depends on its type. Numeric attributes have 10 equal segments (determined empirically) whereas nominal attributes have the same number of segments as they have distinct values.

The illustration of the indexing and retrieval processes is shown in figure 1.

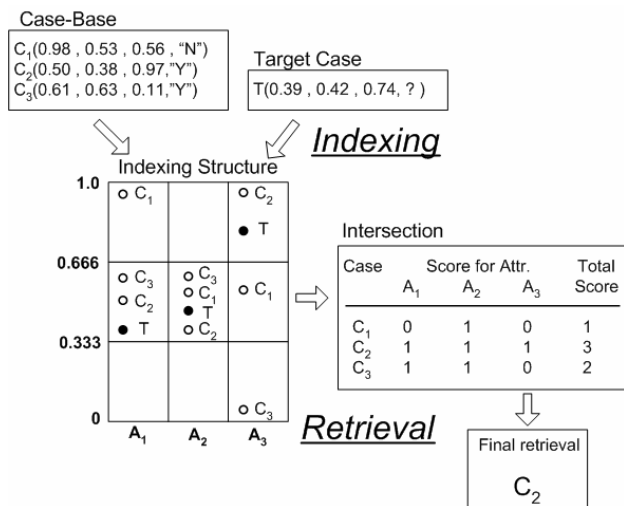


Fig. 1. Illustration of the indexing and retrieval processes for D-HS.

During the indexing process a case is indexed into the matrix based on each attribute individually. The segment, into which the case falls for each attribute, is calculated according to its normalized attribute value. In figure 1 three cases are indexed. The first attribute value of case C₁ is 0.98. Based on the example matrix in the diagram (where only 3 segments are visualized), C₁ is mapped into the 3rd segment bounded by interval [0.666, 1.0) for attribute 1.

Attributes 2 and 3 have values of 0.53 and 0.56 and as such both are mapped into the 2nd segments for their attributes respectively. Cases C₂ and C₃ are similarly indexed. During the retrieval process a target case T is mapped onto the indexing structure in the same way as cases were indexed. All cases whose attribute values fall into the same segments as the target case attribute values are extracted and the degree of intersection (similarity) calculated. This quantifies how many times each case attribute falls into the same segment as the targets. The maximum intersection is equal to the number of attributes. The set of cases with the highest intersection similarity score is selected and used to form a solution. From diagram 1 it can be seen that C₂ is selected as it has the highest similarity.

Promising results from employing D-HS to both classification and regression CBR problems [10] inspired further research into extending and adapting it for T-CBR problems. DFT was chosen as a dimensionality reduction technique, which transformed the temporal attributes into sets of complex frequency coefficients. The workflow for indexing and retrieval of cases with 3 temporal attributes is shown in figure 2.

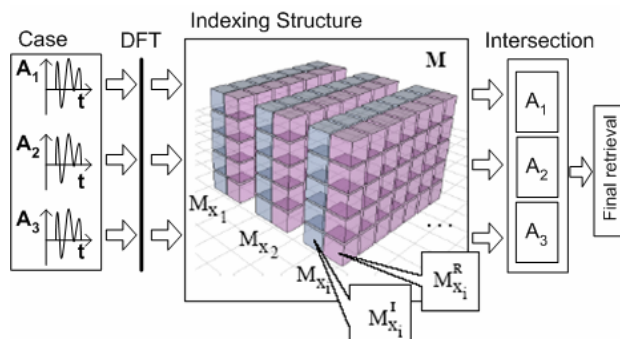


Fig. 2. Illustration of the workflow for indexing and retrieval of cases with 3 temporal attributes.

Each temporal attribute is indexed by the modified D-HS indexing structure M_{x_i} , which is shown in figure 3.

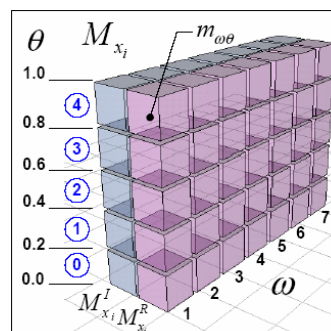


Fig. 3. Illustration of the modified D-HS structure for indexing a temporal attribute.

It is very similar to the D-HS matrix except it has an extra dimension (ω) for indexing complex frequency coefficients for each temporal attribute, generated as a result of DFT. Each frequency coefficient (the columns) consists of two components. The first (column $M_{x_i}^R$) indexes cases based on the real parts of extracted frequency coefficients and the second (column $M_{x_i}^I$) indexes cases based on imaginary parts. Each temporal attribute therefore has both these components. Rows define the number of segments (θ) into which cases are split (as with D-HS usually 10). The mapping of cases into the indexing structure is carried out in the same way as was described for D-HS i.e., sequentially for each temporal attribute. During the retrieval process a temporal query case is mapped onto the indexing structure, and retrieval carried out as before based on the intersection of each individual temporal attribute. Experiments using D-HS^T are presented in the results section, and show promising efficiency and accuracy results in comparison to R-tree.

A detailed analysis of the results showed that some accuracy was being lost by D-HS^T due to poor distribution of values within the frequency coefficients. These distributions often showed that values can sometimes be almost entirely “squeezed” into a single segment. In this situation the process of equal frequency coefficient segmentation does not work. The box-plot in figure 4 highlights this problem where 3 real components of the complex frequency coefficients are shown discretized into 5 segments.

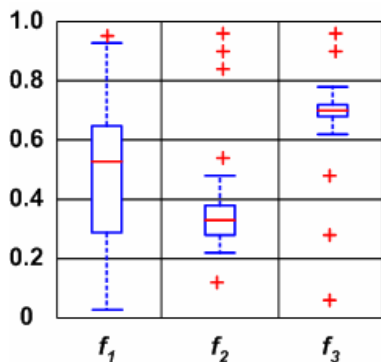


Fig. 4. Distribution of values for the real component of 3 complex frequency coefficients of a temporal attribute.

Values of the first frequency coefficient are split mostly across segments 2, 3 and 4, with some values in all other segments, whereas values for the second and third frequency coefficients are almost entirely subsumed by segments 2 and 4 respectively. Descretizing into equal sized segments works well when the distribution of their values is similar to f_1 . If the distribution is as described by f_2 or f_3 there is little point in equal discretization. It was seen that often with DFT transformed temporal attributes,

the situation as described by f_2 and f_3 was commonplace and this inspired modifications to D-HS^T namely, D-HS^{T(E)} and D-HS^{T(EW)}, to improve its competency.

The difference between these variations of D-HS^T is in the way they define similarity and in the number of segments used in the matrix. D-HS^{T(E)} (E is entropy) is a modification of D-HS^T, where Fayyad and Irani's recursive minimum entropy approach with the Minimum Description Length (MDL) [13] stopping criteria is used in a preprocessing step to produce the optimal number of intervals for each attribute. This addresses the attribute value distribution problem. The second modification, D-HS^{T(EW)} modifies D-HS^{T(E)} to use a weighted intersection procedure.

D-HS^{T(E)} provides an improvement to the indexing process. In D-HS^T the number of intervals was fixed in advance, however, this approach is inflexible and does not provide optimal interval splitting for the indexing matrix, especially when there is an abnormal distribution of transformed attribute values.

The difference between creating a fixed and optimal number of segments is shown in figure 5. The example in figure 1 was taken as a basis for the following explanation, It is important to note that figure 1 was designed to show the indexing of non temporal cases. For the current example in figure 5 we assumed that DFT was applied to three cases C_1, C_2, C_3 each consisting of one temporal attribute. For simplicity we show only the real component of 3 complex frequency coefficients of this temporal attribute. This example provides a clear illustration of the differences between the indexing and retrieval process for D-HS^T and D-HS^{T(E)}

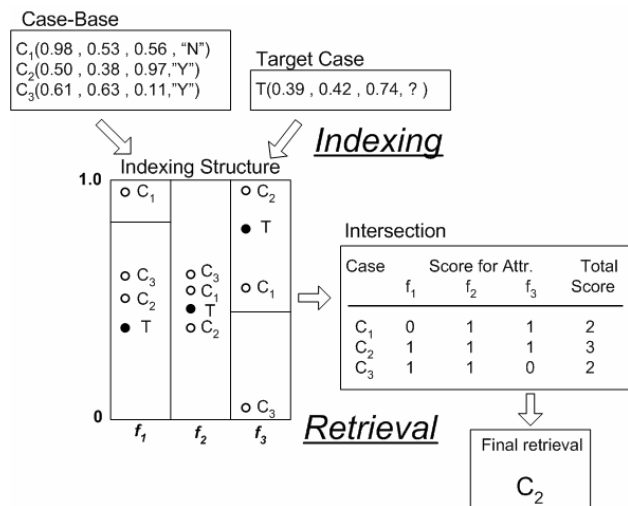


Fig. 5. Illustration of the indexing and retrieval processes for D-HS^{T(E)}.

The preprocessing step defines the optimal number of segments for each frequency coefficient individually. For example two segments were defined for the coefficient f_1 , no segments for the coefficient f_2 and two for f_3 . It can be seen, that segment cut points for coefficient f_1 and f_3 are different, according to the different distribution of their values and their relationship with the class attribute. Apart from discretization all other steps are performed as was described for $D-HS^T$.

$D-HS^{T(EW)}$ was introduced to improve the accuracy of $D-HS^{TE}$ even further. A weak point of $D-HS^{T(E)}$ is the potentially large gap between the target case coefficient values and the coefficient values of cases lying in the same segment. For example, coefficient f_2 in figure 5 was not split. All cases are therefore taken into consideration during the retrieval process based on this coefficient. However it would be advantageous to determine a specific region wherein the closest cases to the target fall.

$D-HS^{T(EW)}$ uses the same procedure as $D-HS^{T(E)}$ for creating the indexing structure and mapping cases onto it. The only change is in the retrieval phase. When segments for the target case coefficients are defined, an intersection procedure which takes into consideration the distance between the target and case coefficient values in each segment is applied. Lets review an example of weighted intersection shown in figure 6.

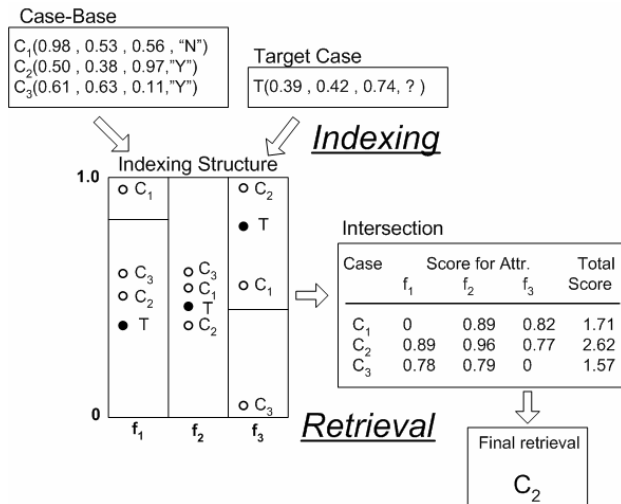


Fig. 6. Illustration of the indexing and retrieval processes for $D-HS^{T(EW)}$.

As can be seen indexing is as before with $D-HS^{TE}$ but during retrieval the weighted intersection, that is the absolute distance between target and case frequency coefficient values, is calculated as the intersection (similarity) criterion. Lets look at the intersection (similarity) score for the case C_1 . As it was not split into the same segment as the target T for attribute 1, its similar-

ity for this attribute is 0. The distance between C_1 and T based on frequency coefficient f_2 is $0.89 (1 - |0.42 - 0.53| = 0.89)$ and for coefficient f_3 $0.82 (1 - |0.74 - 0.56| = 0.82)$ accordingly. The total similarity for case C_1 is 1.71. The same calculation is carried out for the other two cases, which have similarities of 2.62 and 1.57 respectively. The case with the highest similarity is selected, in the current example this is C_2 . This modification improves the retrieval accuracy within a segment but does not solve the problem of the target case lying close to the segment border whereby all nearby cases in the adjoining segment are ignored. However, results show, that this is not a major limitation of proposed modification.

Experimental Technique and Results

Nine temporal case-bases, the majority of those were obtained from Keogh's collection [14], were used in the experiments, where 7 (*asl, ecg, gunx, income, population, temperature* and *rgb*) of them were real world and 2 (*cbf* and *cc*) were synthetic.

In all instances the case-bases were split into training (9/10) and test (1/10) sets. Ten-fold cross validation was carried out and the classification accuracy was noted for each technique along with the associated efficiencies. The significance in competencies for each technique compared to an R-tree was calculated using paired t-tests (significance level 0.05). DFT was chosen as a dimensionality reduction technique. We extracted three frequency coefficients, which were then used in the indexing process.

Three different experiments were carried out, where an R-tree was used as the benchmark against $D-HS^T$, $D-HS^{T(E)}$ and $D-HS^{T(EW)}$ respectively. Experimental results for $D-HS^T$ are shown in figure 7.

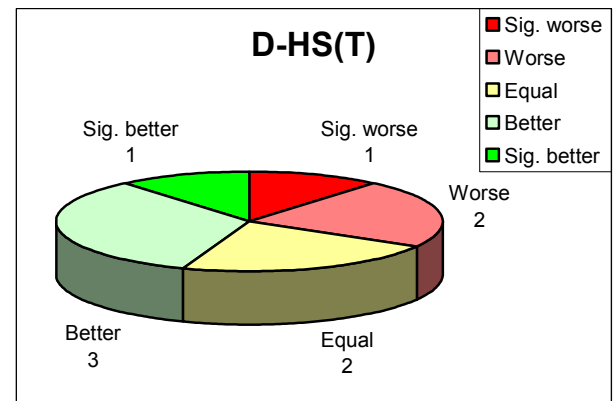


Fig. 7. The distribution of experimental results between $D-HS^T$ and R-tree.

Figure 7 shows the percentage split between five result categories: Those that produced "significantly worse"

results than an R- tree, those that were “worse” but not significantly so, those that were “equal” to an R-tree (not significantly different and less than 1%), those that were “better” but not significantly so and those that were “significantly better” than an R-tree.

It can be seen from figure 7 that the split between “significantly worse” and “significantly better” is equal. One case-base *rgb* showed a “significantly worse” result for D-HS^T compared to the R-tree with nearly a 16% drop in accuracy. Conversely, D-HS^T significantly outperformed the R-Tree with *asl* by nearly 20%. Three case-bases gave better accuracies *gunx* (2.5%), *income* (12.8%), and *population* (18.7%), two case-bases *cbf* and *temperature*, showed “equal” results and the other two, *cc* and *ecg*, showed “worse” results with a 6.6% and 8.3% drop in accuracy respectively. From this it can be concluded that D-HS^T provides the same accuracies as R-tree in T-CBR.

Accuracies improved, after the experiments with D-HS^{T(E)} were carried out. Results are shown in figure 8.

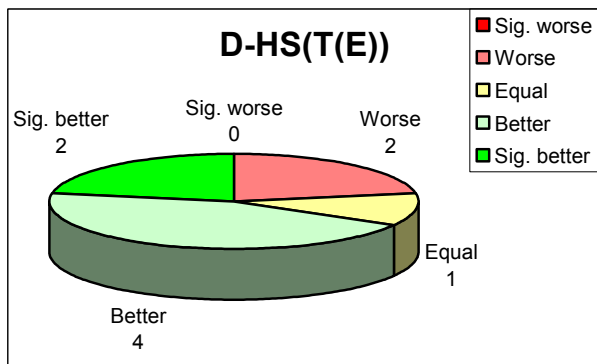


Fig. 8. The distribution of experimental results between D-HS^{T(E)} and R-tree.

Rgb, which had shown a “significantly worse” result with D-HS^T, now provided a “significantly better” result with D-HS^{T(E)} (13.7%). As with D-HS^T, *asl* also provided a significantly better result, but this time by a greater margin than before (33.08%). Those case-bases falling into the “better” category was increased from three to four *cc*, *ecg*, *gunx* and *population* with improvement percentages of 2.0% , 29.2%, 5.5% and 12.5% respectively. The number of “equal” results dropped to one *cbf* and the number of “worse” results remained the same with *income* and *temperature* (2.1% and 3.3%). Overall in comparison to D-HS^T, no case-bases were significantly worse with D-HS^{T(E)} there was 1 extra case-base that was now significantly better and an extra case-base that was better. From this it can be seen that D-HS^{T(E)} outperforms R-tree for T-CBR.

The results for the D-HS^{T(EW)} modification of the D-HS^T technique are shown in figure 9. In general the results distribution for D-HS^{T(EW)} remained practically the same as

for D-HS^{T(E)} apart from one case-base *gunx* which was moved from category “better” to the category “significantly better” with an accuracy improvement of 11.2%. The rest of the case-bases remained in the same categories as with D-HS^{T(E)}. Changes only took place in terms of the percentage accuracy figures compared to R-trees. For two of the “significantly better” case-bases *asl* and *rgb* the outperforming percentage increased slightly to 36.1 and 16.3. For case-bases *cc*, *ecg* and *population* which showed “better” results, the percentage differences were 2.3%, 20.8% and 12.5%. *Cbf* continued to show an “equal” result and *income* and *temperature*, showed a “worse” result as before with percentage differences of 12.8% and 3.3% respectively. As with D-HS^{T(E)}, this variation outperforms R-trees and all other D-HS^T variations.

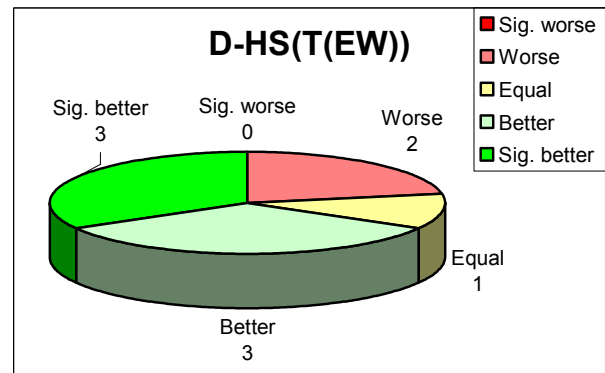


Fig. 9. The distribution of experimental results between D-HS^{T(EW)} and R-tree.

These results are all very encouraging but they must be taken within the context of the efficiencies of the algorithms. The average efficiency ratios for the original D-HS^T and its two modifications compared to an R-Tree are shown in figure 10. A ratio >1 indicates that D-HS^T more efficient and <1 less efficient than R-tree.

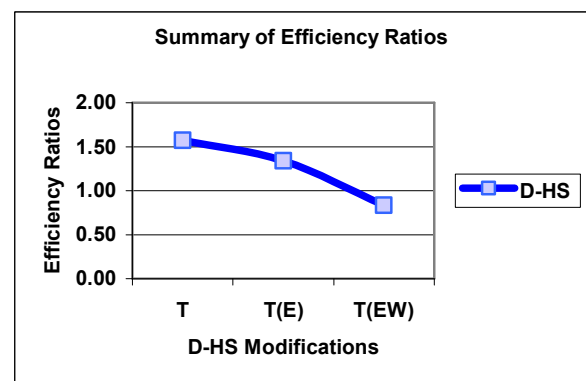


Fig. 10. The summarization of Efficiency Ratios

It is clearly seen from the figure that the efficiency degrades from 1.57 (57% faster than R-Tree) for D-HS^T to 1.34 for D-HS^{T(E)} and finally down to 0.83 (17% slower) for D-HS^{T(EW)}. Such degradation in efficiency is caused by an overhead during the indexing process. The extra step taken by D-HS^{T(E)} for creating the optimal number of intervals is a quite time consuming, due to the large number of calculations required for defining average entropy measure. The further degradation of the efficiency for D-HS^{T(EW)} was caused by the additional extra calculations during the retrieval phase, where instead of calculating a basic similarity score, the more competent algorithm was applied, which takes distances between target and case attributes within segments into consideration.

Conclusions & Future Work

Three variations of the D-HS^T technique for indexing and retrieving temporal cases have been proposed and benchmarked with 9 temporal case-bases for accuracy and efficiency against the commonly used R-tree approach. It was seen that although D-HS^T provided the same accuracies to an R-tree (and better efficiencies) there was a limitation with the basic D-HS^T algorithm. This was due to an observed drop in accuracy with some case-bases, due to a poor distribution of temporal attribute values in the frequency domain. Two modifications were proposed, D-HS^{T(E)} and D-HS^{T(EW)} to address this problem and improve accuracy. Experimental results show that these modifications provided results that were superior in accuracy to R-trees. However the average efficiency ratio, not unexpectedly, showed a slight degradation from 1.57 with D-HS^T to 1.34 with D-HS^{T(E)} and 0.83 for D-HS^{T(EW)}. This was due to the extra overheads involved during the indexing and retrieval processes.

From this we can conclude that D-HS^T and its variations provide a very useful mechanism for indexing and retrieving temporal cases. The variation utilized should depend on the aims and objectives of the user. If speed is the primary requirement then the basic D-HS^T should be chosen as it is faster than R-trees, but equally competent. If maximum accuracy is desired then D-HS^{T(EW)} is the best option. The main focus of the future work is an improving efficiency by introducing a preprocessing step to quickly estimate the optimal interval number from a sample of the data and intelligently predict the most applicable indexing and retrieval approach. It should also be noted that these variations have also been tested on regression case-bases and shown to be equally effective.

References

[1] Dørnum, M., Aamodt, A. and Skalle, P. Representing *Temporal Knowledge for Case-Based Prediction..* (2002). European Conference on CBR, Springer, pp. 174-188.

[2] Combi, C. and Shahar, Y. *Temporal reasoning and temporal data maintenance in medicine: issues and challenges.* Computers in Biology and Medicine (in press).

[3] Workshop “*Applying CBR to Time Series Prediction*”. (2003). Fifth Int'l. Conference. On CBR, pp 213-272.

[4] Rafiei, D. and Mendelzon, A. *Efficient retrieval of similar time sequences using DFT.*(1998) Int'l. Conference on Foundations of Data Organization and Algorithms.

[5] Chan, K. & Fu, A. W. (1999). *Efficient time series matching by wavelets.* Int'l Conference on Data Engineering. pp 126-133.

[6] Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, S. (2001). *Locally adaptive dimensionality reduction for indexing large time series databases.* Conference on Management of Data. pp 151-162.

[7] Agrawal, R., Psaila, G., Wimmers, E. L. and Zait, M. (1995). *Querying shapes of histories.* Int'l Conference on Very Large Databases. pp 502-514.

[8] Mitra, S. and Achary, T. (2003). *Data Mining: Multimedia, Soft Computing and Bioinformatics.* John Wiley & Sons. pp 343-344.

[9] Guttman, A. *R-trees: a dynamic index structure for spatial searching.* (1984). Int'l Conference on Management of Data. pp. 47-57.

[10] Patterson, D., Rooney, N. & Galushka, M. (2002) *Efficient Similarity Determination and Case Construction Techniques For Case-Based Reasoning.* European Conference on CBR. pp. 292-305.

[11] Patterson, D., Rooney, N. & Galushka, M. (2004). *An Effective Indexing and Retrieval Approach for temporal cases.* Proceedings of the 17th International FLAIRS Conference. AAAI Press.

[12] Patterson, D., Rooney, N. & Galushka, M. (2003). *Efficient Real Time Maintenance of Retrieval Knowledge in Case-Based Reasoning.* Proceedings of the 5th International Conference on Case Based Reasoning. Trondheim, Norway Springer LNAI.

[13] Fayyad, U. & Irani, K. (1993) *Multi-interval discretization of continuous-valued attributes for classification learning.* In Proceedings of the 13th Int. Joint Conference on Artificial Intelligence.

[14] Keogh, E. & Folias, T. (2002). The UCR Time Series Data Mining Archive [<http://www.cs.ucr.edu/~eamonn/TSDMA/index.html>]. Riverside CA. University of California - Computer Science & Engineering Department